

LISTING OF THE CLAIMS

At the time of the Action:

Pending Claims: 1-13, 15-23, 25-29, 31-38 and 40

Withdrawn Claims: none

Canceled Claims: 14, 24, 30, 39 and 41-74

After this Response:

Pending Claims: 1-13, 15-23, 25-29, 31-38 and 40.

Amended Claims: 1 and 40.

Canceled Claims: 14, 24, 30, 39 and 41-74.

1. (Currently Amended) Interfaces, stored on one or more computer-readable media, to be called on kernel transaction management objects, comprising:

application program interfaces (APIs) local with the transaction manager located in a kernel to implement operations in the kernel on a kernel transaction object (TX), the TX representing a transaction, the TX and being accessible by at least one process participating in the transaction;

APIs local with the transaction manager to implement kernel-level operations on a kernel resource management object (RMO), the RMO representing a relationship between a TX associated with the transaction manager and at least one resource that participates in the transaction, the resource being an entity capable of storing data in a durable state; and

APIs local with the transaction manager to implement kernel-level operations on a kernel enlistment (EN) object, the EN representing a relationship between a resource manager and the transaction.

2. (Previously Presented) Interfaces according to Claim 1, wherein each of the APIs to implement operations on the TX, the RMO, and the EN utilize a handle to refer to an object.

3. (Original) Interfaces according to Claim 2, wherein each of the handles is an opaque reference to a unique object.

4. (Previously Presented) Interfaces according to Claim 2, wherein at least one of the APIs calls for the TX to transmit pre-prepare messages to resource managers associated with a transaction.

5. (Previously Presented) Interfaces according to Claim 2, wherein at least one of the APIs calls for the TX to transmit a prepare request to resource managers enlisted in a transaction.

6. (Original) Interfaces according to Claim 2, wherein at least one of the APIs calls for a new TX to be created for a transaction.

7. (Previously Presented) Interfaces according to Claim 2, wherein at least one of the APIs calls for the TX to be opened for a transaction.

8. (Previously Presented) Interfaces according to Claim 2, wherein at least one of the APIs calls for the TX to commit a transaction.

9. (Previously Presented) Interfaces according to Claim 2, wherein at least one of the APIs calls for the TX to abort a transaction.

10. (Previously Presented) Interfaces according to Claim 2, wherein at least one of the APIs calls for the TX to save a current state of the transaction.

11. (Previously Presented) Interfaces according to Claim 2, wherein at least one of the APIs calls for the TX to retrieve information about the TX for a requestor.

12. (Previously Presented) Interfaces according to Claim 2, wherein at least one of the APIs calls for the TX to set information.

13. (Previously Presented) Interfaces according to Claim 2, wherein at least one of the APIs calls for the TX to close.

14. (Canceled).

15. (Previously Presented) Interfaces according to Claim 2, wherein at least one of the APIs calls for a new RMO to be created.

16. (Original) Interfaces according to Claim 15, wherein the new RMO is volatile.

17. (Original) Interfaces according to Claim 15, wherein the new RMO is durable.

18. (Previously Presented) Interfaces according to Claim 2, wherein at least one of the APIs calls for the RMO to open for a transaction.

19. (Previously Presented) Interfaces according to Claim 2, wherein at least one of the APIs calls for the RMO to be destroyed.

20. (Previously Presented) Interfaces according to Claim 2, wherein at least one of the APIs calls for the RMO to transmit information regarding the RMO to a requestor.

21. (Previously Presented) Interfaces according to Claim 2, wherein at least one of the APIs calls for the RMO to set information.

22. (Previously Presented) Interfaces according to Claim 2, wherein at least one of the APIs calls for the RMO to be enlisted on a transaction at least once.

23. (Previously Presented) Interfaces according to Claim 2, wherein at least one of the APIs calls for a notification from a resource manager for the RMO.

24. (Canceled).

25. (Previously Presented) Interfaces according to Claim 2, wherein at least one of the APIs is to implement operations on the TX by the RMO.

26. (Previously Presented) Interfaces according to Claim 25, wherein the at least one of the APIs is to inform the TX that pre-preparing is complete.

27. (Previously Presented) Interfaces according to Claim 25, wherein the at least one of the APIs is to inform the TX that transaction preparation has been completed by a requested resource manager.

28. (Previously Presented) Interfaces according to Claim 25, wherein the at least one of the APIs is to inform the TX that a resource manager has completed rolling back a transaction.

29. (Previously Presented) Interfaces according to Claim 25, wherein the at least one of the APIs is to inform the TX that a resource manager has committed to a transaction.

30. (Canceled).

31. (Previously Presented) Interfaces according to Claim 2, wherein at least one of the APIs calls for a resource manager to be registered as a communications resource manager for a particular protocol.

32. (Previously Presented) Interfaces according to Claim 2, wherein at least one of the APIs calls for a representation of a transaction to be serialized into a buffer.

33. (Previously Presented) Interfaces according to Claim 2, wherein at least one of the APIs calls for information representing registered protocols to be serialized into a buffer.

34. (Previously Presented) Interfaces according to Claim 32, wherein at least one of the APIs calls for a transaction represented by the serialization be made available by a transaction management object.

35. (Previously Presented) Interfaces according to Claim 2, wherein at least one of the APIs calls for a transaction to be propagated to a destination using push-style propagation.

36. (Previously Presented) Interfaces according to Claim 35, wherein at least one of the APIs calls for the output of the API calls for the transaction to be propagated to a destination using push-style propagation to be retrieved.

37. (Previously Presented) Interfaces according to Claim 31, wherein at least one of the APIs is called when transaction propagation has been completed.

38. (Previously Presented) Interfaces according to Claim 31, wherein at least one of the APIs is called when a requested transaction propagation has failed.

39. (Canceled).

40. (Currently Amended) An apparatus for implementing a transaction, comprising:

a kernel transaction object (TX) to represent a transaction, the TX being accessible by at least one process participating in the transaction;

a kernel resource manager object (RMO) to represent a relationship between a TX associated with the transaction manager and at least one resource

that participates in the transaction, the resource being an entity capable of storing data in a durable state; and

a kernel enlistment object (EN) to represent a relationship between a resource manager and the transaction,

wherein two-phase commit processing is executed at the kernel-level by calling application program interfaces (APIs) on the TX, the RMO, and the EN, the APIs local with the transaction manager, the transaction manager located in a kernel of an operating system.

41-74 (Canceled).